CLEI Electronic Journal, Volume 28, Issue 1, paper 8, February 2025

Yogasana classification using Deep Neural Network : A Unique Approach

Rishi Raj[1], Minakshi Sarkar[2], Rajesh Mukherjee[3], Bidesh Chakraborty[4*]

[1,2]Computer Science & Engineering (Data Science), Haldia Institute of Technology, ICARE Complex, Haldia, 721657, West Bengal, India.
[3]Computer Science & Engineering, Haldia Institute of Technology, ICARE Complex, Haldia, 721657, West Bengal, India.
[4*]Computer Science & Engineering (AIML), Haldia Institute of Technology, ICARE Complex, Haldia, 721657, West Bengal, India.

*Corresponding author(s). E-mail(s): bidesh.me@hithaldia.ac.in;
Contributing authors: iamrishiraj31@gmail.com;
minakshi.sarkar42@gmail.com; rajeshmukherjee.cse@gmail.com;

Abstract

Yogasana, often simply referred to as "asana", is a Sanskrit term that encompasses the physical postures or poses practiced in yoga. Traditionally, there are almost 90 classical Yogasanas, each with its unique benefits and variations. It's very important to identify the postures and benefits of each asana. Classifications of Yogasana can have a significant impact on the practice and promotion of yoga, making it more accessible, educational, and safe for people of all skill levels. In this research work, we have automatically identified different Yogasanas from images using a deep neural network. For our work we have considered 10 very popular categories of Yogasana and built a dataset of volume 800 images. Furthermore, the data augmentation (DA) technique is utilized to increase the volume of data by including slightly modified replicas of existing data. Xception, combined with a self attention network, has been employed to classify various Yogasanas based on diverse images. The model attains an impressive validation accuracy of 99%, surpassing the performance of all other contemporary state-of-the-art models.

Keywords: Yogasana Classification, CNN, Xception, Attention Network

1 Introduction

Yoga has grown in popularity among medical professionals. Yoga has several advantages, including enhanced health, mental strength, weight loss, and so on. The research [1, 2]outlines the healing benefit of yoga and gives a thorough evaluation of the advantages of regular yoga practice.

However, Yoga should only be performed under the guidance of a qualified practitioner, as improper or misalignment can result in health issues such as strains in the ankles, stiff necks, muscular pulls, and so on. Yet, in the contemporary setting, people find greater comfort within their own homes, engaging in nearly all activities through online means. This condition necessitates a technologically based yoga practice. This may be performed via smartphone apps or virtual tutoring software [3]. In this research, we provide a classification model to recognize and identify yogasana classes from an image.

The task of image classification using deep learning has become significantly more accessible due to the widespread availability of hardware resources. Various cloud platforms, such as Kaggle Notebooks or Google Colab, offer researchers the to write programs and access GPU services seamlessly. With the ease of hardware availability, deep learning has found applications in diverse fields, including medical science [4], agriculture [5], and the finance sector [6].

Convolutional Neural Networks (CNNs) represent a powerful class of deep learning (DL) models specifically designed for processing and analyzing visual data [7]. CNNs have revolutionized image recognition tasks by automatically learning hierarchical features through convolutional (CONV), pooling, and fully connected (FC) layers. These networks excel at capturing intricate patterns and spatial relationships within images, making them highly effective for image classification [8], object detection [9], and segmentation tasks [4].

Pre-trained models are a key advancement in the realm of deep learning. These models, trained on massive datasets like ImageNet, serve as valuable starting points for various computer vision applications. By leveraging the knowledge gained from extensive pre-training, these models can be fine-tuned on smaller datasets, offering impressive performance even with limited labeled examples. Popular pre-trained models such as ResNet [10], DensNet [11], Inception [12], and Xception [13] provide a wealth of learned features, enabling researchers and developers to transfer this knowledge to new tasks efficiently. This transfer learning approach significantly accelerates model training and enhances performance across a range of visual recognition tasks.

In our research endeavor, we initially curated a comprehensive Yagasana dataset, encompassing nearly 800 images across ten distinct classes. Sample images from the dataset are depicted in Figure 1. Figure 2 depicts the balanced and uniform distribution of our dataset across ten classes. Subsequently, we devised a sophisticated deep learning image classification model that integrates the Xception architecture with a Self-Attention Network. Unlike existing methods that primarily rely on conventional convolutional neural networks (CNNs), our approach leverages the strength of Xception's depthwise separable convolutions alongside self-attention to capture both spatial and contextual dependencies more effectively.Our proposed method achieves significant improvements in classification accuracy and robustness compared to traditional

models by dynamically attending to important features within Yogasana pose images. This approach effectively reduces irrelevant information while enhancing pose distinction, enabling more precise and reliable classification. Unlike previous studies that primarily relied on conventional deep learning architectures without attention mechanisms, our work introduces a self-attention-enhanced Xception model. This integration allows for superior performance by capturing both spatial and contextual dependencies, resulting in improved generalization across diverse Yogasana poses. The proposed methodology demonstrated efficacy in effectively assimilating information from the target domain, with a particular emphasis on crucial features in significant regions. This enhancement substantially improved the model's ability for feature extraction.

This study makes noteworthy contributions in three key aspects:

1. The establishment of a Yagasana dataset comprising close to 800 images spanning ten diverse classes.
2. The optimization of the network structure was achieved by iteratively reducing convolutional layers while monitoring recognition accuracy, building upon existing network architectures.
3. The integration of Xception with a Self-Attention Network, a novel approach that enables the model to focus on pertinent regions within input images, thereby augmenting its capacity to discern intricate patterns and relationships. The model achieves an outstanding validation accuracy of 99%, outperforming all other current state-of-the-art models.

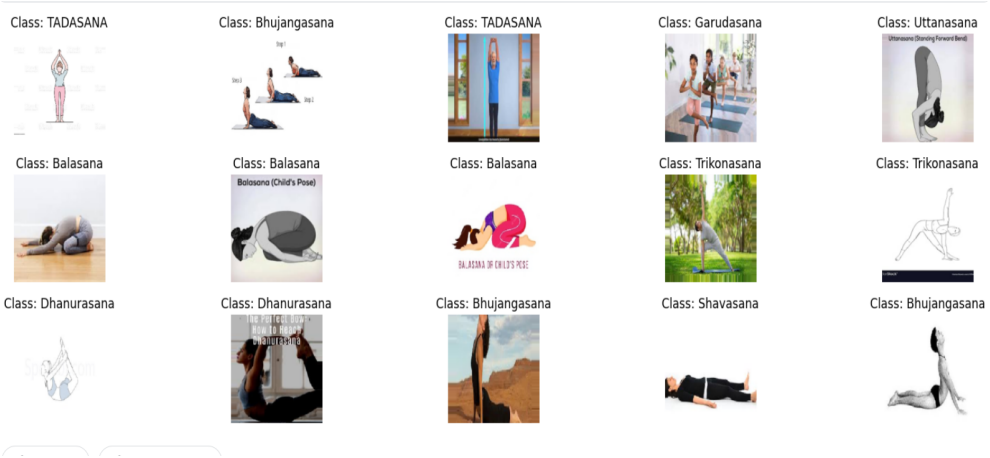

Fig. 1: Random sample images from Dataset

2 Literature review

Yoga, the subject of this study, is not frequently explored, hence just a few publications on the subject have been unearthed.

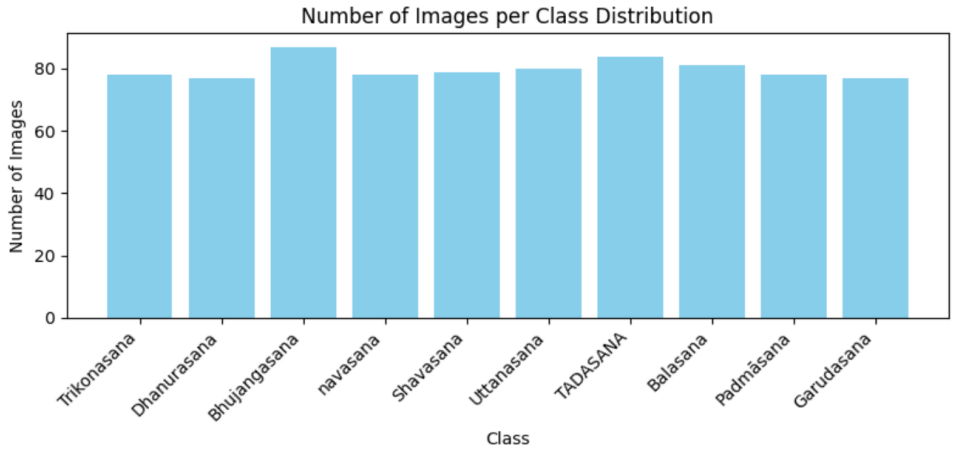

Fig. 2: Image distributions from the Dataset

The classification model [14] was developed by merging deep learning [15] with image processing The whole work is dedicated to categorising 10 distinct types of yogasanas. The DL model evaluated and detected all the postures and was then used to establish the precise class of the yoga pose.

Sruti Kothari [16] presented a computational approach utilizing CNN for identifying yoga poses from images. Their classification model was trained on a dataset comprising 1000 images categorized into six classes, achieving an accuracy of approximately 85%. Meanwhile, Hua-Tsung Chen [17] developed a system for recognizing a yoga trainer's poses. Initially, a kinetic method was employed to capture the user's body map and extract contours. Following this, a star skeleton was generated using a swift skeletonization method that connects the centroid to different joint parts. This approach demonstrated a high accuracy rate of 96%.

Edwin W.trejo [18] developed a model for yoga position correction. An expert will provide real-time instruction to the user on the posture adjustment. They created a comprehensive database for calculating 6 yoga positions using an identification technique centred on the AdaBoost algorithm. Finally, the accuracy is 94.78%.

Further, a highly computationally efficient solution for real-world yoga position detection is described in [19]. A 3D CNN architecture is employed to recognize yoga positions in real time. Fully linked computational layers are pruned for computational efficiency, and supplemental layers such as batch normalization (BN) and average pooling are implemented. As a consequence, 91.5% accuracy is reached.

A dataset with six yoga asanas is produced in [20]. CNN is employed in this study to extract features from critical locations, while LSTM is used to provide temporal prediction. As a consequence, 99.04% accuracy is acquired. An automated process for annotating dance films based on the position of the foot is presented in [21, 22]. They employed transfer learning (TL) to do this. TLis utilised to extract features from images, and a deep neural network is employed for image categorization. The achieved accuracy was encouraging.

Matthias Dantone describes a new, joint regression approach [23] for determining human locations from 2D images. Two layered random forests were employed as joint regressors. The initial layer employed a discriminative independent body component classifier, while the next layer predicted joint locations by simulating the parts' interdependence and co-occurrence.

Sadeka Haque [24] suggested a CNN based novel method to estimate human postures from a 2D human activity images. To support this, they provided a dataset comprising 2000 images representing five distinct types of exercises and after a series of trials, they reached an accuracy of 82.68%.

Dushyant Mehta [25] created an approach for 3D motion capture using an RGB camera. Using CNN to estimate 2D and 3D posture parameters, as well as identify all visible joints in people. To improve information flow while retaining accuracy, he created a revolutionary architecture known as SelecSLS Net. The second phase involves converting each individual's postural information into a comprehensive 3D pose estimation using a fully linked neural network. In the third stage, a space-time skeleton model is fitted to the predicted model, yielding a complete skeletal posture in joint angles for each person.

Josvin Jose [26] presented a system that uses DL methods like CNN and TL to identify a yoga pose from an image or a frame of a movie. They trained the system and assessed its prediction accuracy using pictures of 10 different asanas. With the use of TL, the prediction model is able to attain 85% prediction accuracy.

The study [27] explores yoga pose classification using deep learning, emphasizing the impact of skeletonization through the MediaPipe library for body keypoint detection. Various models were evaluated with and without skeletonization, demonstrating that skeletonized images significantly improve classification accuracy. The proposed YogaConvo2d model achieved the highest validation accuracy of 95.06% with skeletonized images, highlighting the effectiveness of this approach. A real-time yoga pose annotation and classification approach using a time-distributed CNN has been studied by [28], demonstrating superior accuracy in yoga pose recognition compared to traditional SVM and KNN classifiers.

3 Methology

This section encompasses a description of the dataset, data augmentation techniques, the fundamentals of CNN and pre-trained models, and our proposed model. The architecture of our model, which is constructed using Xception and Self Attention Network, is further elucidated.

3.1 Dataset Description

We use web scraping techniques to generate our own dataset for our research. There are ten classes of different yoga positions in the dataset. The dataset has been successfully uploaded to Kaggle [1]. The quantities of images belonging to the categories Balasana, Bhujangasana, Dhanurasana, Garudasana, Navasana, Padmasana, Shavasana, Tadasana, Trikonasana, and Uttanasana are 81, 87, 77, 76, 78, 78, 79, 84, 78, and 80,

[1] https://www.kaggle.com/datasets/aishanikaggle/yogasan-dataset

respectively. The dataset contains approximately 800 images. Figure 2 illustrates that our dataset is balanced and evenly distributed across ten classes. Figure 1 shows the random images from our dataset.

3.2 DA Techniques

DA is widely used to artificially enhance training data by making modifications to existing data such as random rotation, shift, zoom, flip, and dividing. We used the ImageDataGenerator from the Keras package, which is often used for image data preparation during neural network training. The data was supplemented in accordance with the Table 1.

Data Augmentation parameter	Value
Rescale	1/255
Rotation Range	30
Zoom Range	0.2
Vertical flip	True
Horizontal Flip	True
Validation Split	0.15

Table 1: Data Augmentation Parameters

3.3 CNN and Pretrain Models

CNN, a deep learning architecture, excels in addressing image categorization challenges by employing convolution and pooling techniques [26, 27]. The fundamental building blocks of CNN include the CONV layer, activation function, pooling layer, and FC layer [28]. The CONV layer acts as a filter on an image, providing a description of all observed pixels. The activation function introduces non-linearity, and the pooling layer facilitates downsampling, reducing the framework's size. In a typical structure, FC layers process a one-dimensional vector, ultimately selecting the classification label with the highest probability of approval.

A pretrained model is a neural network model that has been trained on a large dataset for a specific task before being saved or distributed for use in other related tasks or applications. Architectures such as ResNet[10], DenseNet[11], Inception[12], and Xception[13] are examples of popular pretrained models. Pretrained models can save time, resources, and computational power, making them an invaluable tool in the creation of a wide range of machine learning applications.

In 2014, Google introduced Inception with the objective of improving the training efficiency of large networks [12]. It processes input and effectively reduces the number of parameters by incorporating 1×1 CONV kernels after 3x3 and 5×5 CONV kernels, along with 3×3 pooling kernels. This approach broadens the network and enhances its scalability.

In 2015, ResNet[10], another pretrained model, was introduced to address the issue of vanishing gradients in deep networks. It enables deep networks to effectively learn high-dimensional image attributes by incorporating shortcut connections that merge the outputs of previous layers with the output of the current layer.

Derived from ResNet, DenseNet [11] gives greater emphasis on connectivity strategy, with each layer transmitting its output to all subsequent layers and concatenating the inputs from all preceding layers. This approach enhances the transfer of image features between layers, enabling the model to utilize features more effectively for image classification.

Xception [13] is an improvement on Inception as it uses model parameters more efficiently for the same number of parameters and incorporates residual structure and depthwise separable convolution. In our experiment, we have used Xception with attention mechanism to implement our model. The Xception architecture is explaned in section 3.4.1.

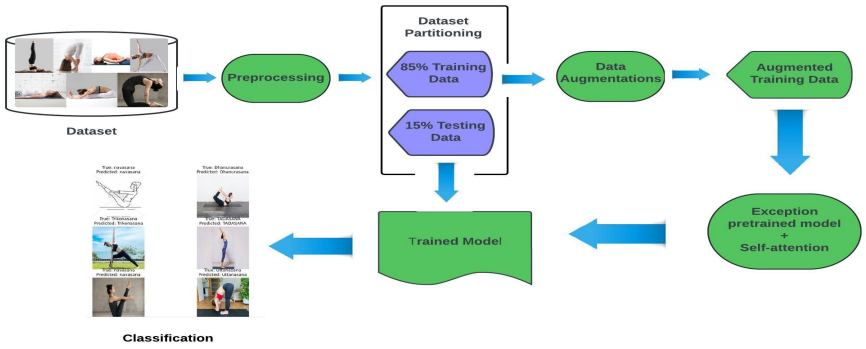

Fig. 3: The system work flow

3.4 Proposed model

The stages needed in training a deep learning model are depicted in Figure 3.The procedure begins with a dataset. Preprocessing is then applied to the data. In machine learning, data preprocessing refers to the practice of preparing (cleaning and organising) raw data for use in developing and training machine learning models. Following then, the dataset is divided into training and testing sets, with data augmentation applied only to the training set. This process involves modifying existing data points to artificially expand the dataset, enhancing its diversity and improving model generalization. After that, the preprocessed data is utilised to build a machine-learning model. To fit into the whole process, two specialised models, "Xception" [13] and "Self attention network" [29], are employed. The trained model is subsequently trained on a training dataset. This stage aids in evaluating the model's performance.
We have applied the self-attention mechanism after the Xception model, allowing it to refine and enhance the extracted features by capturing long-range dependencies

and improving the model's focus on important spatial patterns. This integration contributes to better feature representation, ultimately enhancing classification accuracy.

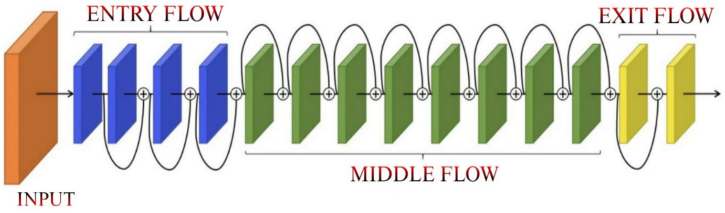

Fig. 4: Xception architecture

3.4.1 Xception Architecture

Figure 4 depicts the Xception architecture. [30]Xception outperforms Inception by including depthwise separable convolution (DSC) and residual structure, optimizing the use of model parameters with an equivalent parameter count. Therefore, in our paper, Xception is predominantly employed to implement our model. The Xception module consists of three flow structures, each incorporating BN, ReLu activation functions, and a 3x3 DSC kernel. The data undergoes processing in the Xception module through the entry flow, followed by the middle flow, and the exit flow. The 36 CONV layers in the architecture are organized into 14 modules. All the modules have linear residual connections, except for the initial and concluding modules.

To address the issue of too many CONV layers hindering effective forward propagation, BN is used to standardise the eigenvalues of each layer, balancing the distribution of the output. ReLU is commonly employed as an activation function in neural network models to introduce non-linearity, accelerate convergence, address the vanishing gradient problem, induce sparsity, and facilitate ease of interpretation. DSC with residual connections in a linear stack simplifies the model structure's definition and modification.

3.4.2 Attention NetWork

The Self-Attention method can assist the model in focusing on more important regions within the imagery, resulting in improved performance for classification tasks with fewer data samples or more complicated image backgrounds [31]. The attention mechanism enables models to learn deeper connections between things and aids in the discovery of fascinating new patterns in data. It also aids in the modelling of long-term, multi-level relationships across distinct image areas. An attention function processes a query and a set of key-value pairs, all in vector form, to generate an output. The output is computed as a weighted sum of the values, with each value's weight determined by the compatibility function between the query and the corresponding key [29].

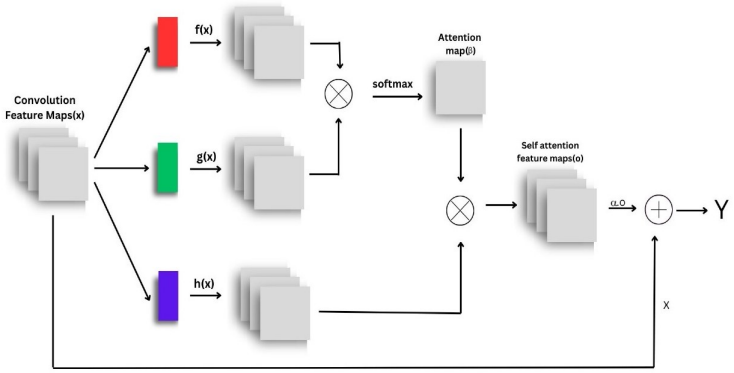

Fig. 5: Self Attention Network

Figure 5 depicts our suggested model. Let $x \in \mathbb{R}^{C\times N}$ be the extracted feature matrix from the xception pretrained model, which is subsequently input into a self-aware module. Consider three feature extractors: f(x), g(x), and h(x). Both h(x) and the input have the same number of channels (C) We discovered that 512 channels provide the highest performance accuracy. Position modules f(x) and g(x) are used to calculate attention. At the i^{th} and j^{th} places, $f(x_i)$ and $g(x_j)$ accept input feature maps. When compared to h(x), $f(x_i)$ and $g(x_j)$ both have fewer channels (C/8). This enables us to exclude noisy input channels and concentrate solely on features crucial to the attention mechanism. We employ 2-D non-strided1-1 CONV and a non-local method within the attention module which, is defined as follows:

$$\eta_{ij}(x) = f(x_i)^T g(x_j) \tag{1}$$

$f(x_i) = W_f x_i$, $g(x_i) = W_g x_i$, where $W_g \in \mathbb{R}^{C\times C}$ and $W_f \in \mathbb{R}^{C\times C}$ are the weight matrices that have been learned. Next, we calculate the softmax of η_{ij} to derive the resulting attention map β_{ij}.

$$\beta_{ij} = \frac{\exp(\eta_{ij})}{\sum_{i=1}^{N} \exp(\eta_{ij})} \tag{2}$$

To acquire the ultimate Self-Attention map, denoted as $o \in \mathbb{R}^{C\times N}$ we perform matrix multiplication between β_{ij} and $h(x_i)$,

$$o_j = \sum_{i=1}^{N} \beta_{ij} h(x_i) \tag{3}$$

Here, $h(x_i)$ is expressed as $W_h x_i$, representing the third input feature channel with C (see Figure 5) and W_h is a learnt weight matrix like W_f and W_g. The Self-Attention mechanism computes a weighted sum across all extracted features of $h(x_i)$ and excludes those with minimal influence as a consequence of this matrix multiplication

step. As a result, the Self-Awareness layer's ultimate output is

$$y_i = \alpha o_i + x_i \tag{4}$$

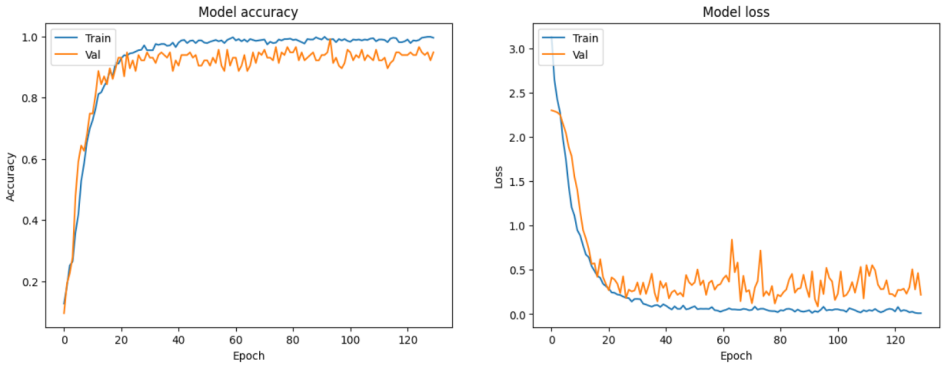

Fig. 6: Model Accuracy and Model Loss

4 Experimental Results

The experiments were carried out using the Kaggle notebook, utilizing a GPU from NVIDIA's Tesla series, specifically the P100 model with 16GB of GPU memory and 29GB of RAM. The experiment was performed on a dataset containing 798 distinct images representing 10 Yogasana classes. The dataset was partitioned into training and testing sets in an 85%:15% ratio. In particular, 678 images were utilized for training, and 120 images were reserved for testing. A deep learning model, designed with an Exception and Self Attention Network, was employed for the classification of the 10 classes of Yogasana poses. We trained our neural network using the Adam optimizer with a learning rate of 0.001. The model was configured to minimize the categorical crossentropy loss function. Training data was processed in batches of 16 samples during each iteration.

The Figure 6 shows two graphs, one for model accuracy and one for model loss. Both graphs appear to show a similar trend, with the model's accuracy increasing and loss decreasing as the number of epochs increases. This suggests that the model is learning over time and improving its performance on the training data. The accuracy graph shows that the model's accuracy on the training data is initially around 0.1, and it increases to around 1.0 after 120 epochs. The loss graph shows that the model's loss on the training data is initially around 3.0, and it decreases to around 0.0 after 120 epochs.

Confusion Matrix:- The confusion matrix presented in Figure 7 is employed for evaluating the performance of our proposed model in Yogasana classification. This matrix, with dimensions 10 x 10 to accommodate the 10 distinct Yogasana

Model Network	LOSS	ACC	F1 Score	PRECISION	RECALL
CNN	0.28	0.61	0.35	0.21	0.99
DenseNet121	0.17	0.90	0.93	0.95	0.93
InceptionV3	1.92	0.42	0.59	0.54	0.65
MobileNet	0.12	0.93	0.96	0.97	0.95
ResNet50	1.72	0.51	0.24	0.62	0.15
Xception	0.13	0.93	0.95	0.96	0.95
DenseNet121 + CBAM	0.80	0.86	0.92	0.94	0.90
InceptionV3 + CBAM	0.47	0.79	0.92	0.94	0.91
MobileNet + CBAM	0.39	0.90	0.98	0.98	0.98
ResNet50 + CBAM	1.5	0.57	0.42	0.69	0.30
Xception + CBAM	0.53	0.78	0.96	0.97	0.95
DenseNet121 + Attention	0.53	0.79	0.82	0.86	0.79
InceptionV3 + Attention	0.52	0.80	0.84	0.90	0.79
MobileNet + Attention	1.01	0.72	0.58	0.66	0.51
ResNet50 + Attention	1.42	0.56	0.41	0.67	0.30
Xception + Attention	**0.02**	**0.99**	**0.98**	**0.99**	**0.98**

Table 2: Performance of pre-trained deep learning models using validation data

classes, serves as a comparison between actual and predicted values. Correct classifications are indicated by the diagonal values, while incorrect classifications are represented by non-diagonal values. Figure 7 reveals that all test images from six Yogasana classes—Balasana, Dhanurasana, Navasana, Shavasana, Trikonasana, and Uttanasana—are accurately predicted. However, misclassifications are observed, with one image each from Bhujangasana and Tadasana, as well as two images each from Garudasana and Padmasana.

The evaluation of the proposed model's performance included the consideration of various metrics such as Accuracy (AC), Precision (PR), Recall (RC), and F1-Score. The formulation of these evaluation metrics is outlined as follows:

$$AC = \frac{(K + M)}{(K + M + N + P)} \tag{5}$$

$$PR = \frac{K}{(K + N)} \tag{6}$$

$$RC = \frac{K}{(K + P)} \tag{7}$$

$$F1 - Score = \frac{2 \times (PR \times RC)}{(PR + RC)} \tag{8}$$

The terms K, M, N, and P represent true positive, true negative, false positive, and false negative, respectively.

ROC Curve:- The ROC (Receiver Operating Characteristic) curve serves as a graphical representation of a classification model's performance across various threshold levels. Figure 8 illustrates the ROC curve for our model. To facilitate explanation,

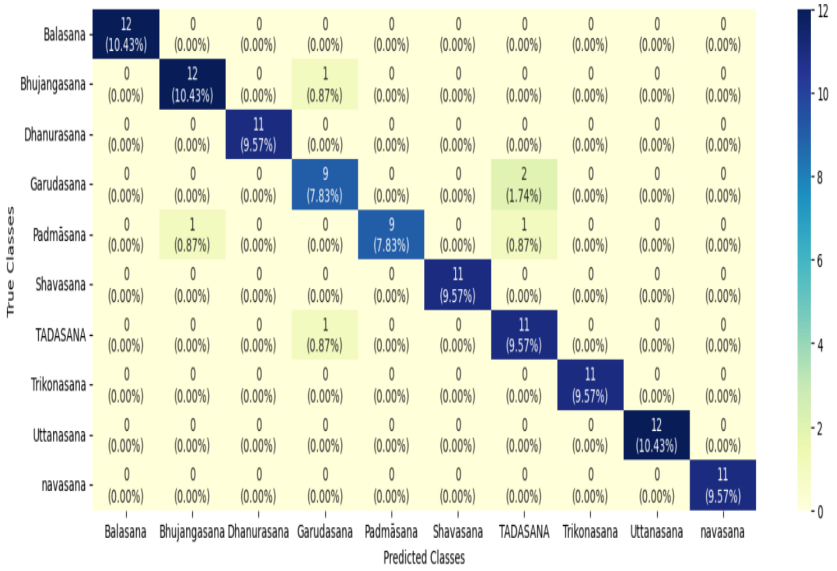

Fig. 7: Confusion Matrix

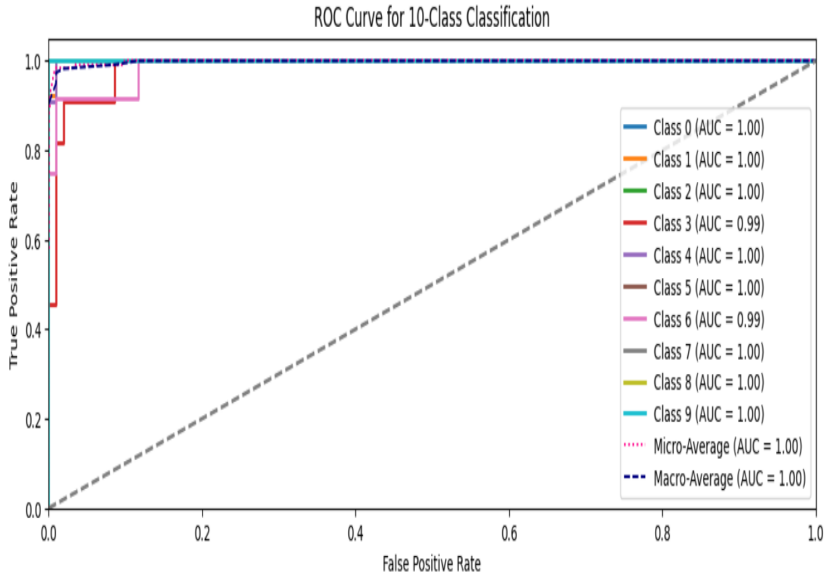

Fig. 8: ROC Curve

we've assigned unique numerical mappings to all Yogasana classes. For instance, Balasana is associated with class 0, Bhujangasana with class 1, and so forth. The mappings continue with Dhanurasana, Garudasana, Padmasana, Shavasana, Tadasana, Trikonasana, Uttanasana, and Navasana being assigned to classes 2, 3, 4, 5, 6, 7, 8, and 9, respectively.

The AUC (area under the curve) values for all classes are very high, ranging from 0.99 to 1.00. This suggests that the model is performing very well at distinguishing between the different classes. The curves for most classes are close to the top left corner of the graph, which is the ideal location for an ROC curve. This means that the model is able to correctly classify a high proportion of positive examples (true positive rate) while still maintaining a low rate of false positives (false positive rate). There are a few curves that are slightly further away from the top left corner, such as Class 3 and Class 6. This could indicate that the model is having a little more difficulty distinguishing between these classes and other classes. An ROC AUC score of 0.9973 is remarkably high. It indicates that the model has an exceptional ability to distinguish between the positive and negative classes.

Existing Work	Dataset Volume	Classes	Accuracy
Sruti Kothari [16]	1000	6	85%
Edwin W.trejo [18]	1200	6	94.78%
Josvin Jose [26]	700	10	85%
Sadeka Haque [24]	2000	5	82.68%
Garg [27]	1551	5	95.06%
Proposed Model	**798**	**10**	**99%**

Table 3: Comparison with existing works

Table 2 presents a comparison of the performance of various CNN models for image classification. In our experiments, we employed five pre-trained models, namely DensNet121, InceptionV3, ResNet50, MobileNet, and Xception. Additionally, the Convolutional Block Attention Module (CBAM) was applied to all these pre-trained models to calculate accuracy. Notably, the most promising results were achieved when the model was constructed with Xception and integrated with the Self Attention Network. Our proposed model, featuring Xception with Self Attention Network, demonstrated exceptional performance, attaining a validation accuracy of 99% and surpassing all the state-of-the-art models outlined in Table 2.

The Xception architecture, with its depthwise separable convolutions, efficiently extracts fine-grained spatial features while reducing computational complexity. By incorporating self-attention after the Xception model, we enhance the model's ability to capture long-range dependencies and focus on the most discriminative features within Yogasana poses. This combination allows for better feature representation, improved pose differentiation, and ultimately superior classification performance.

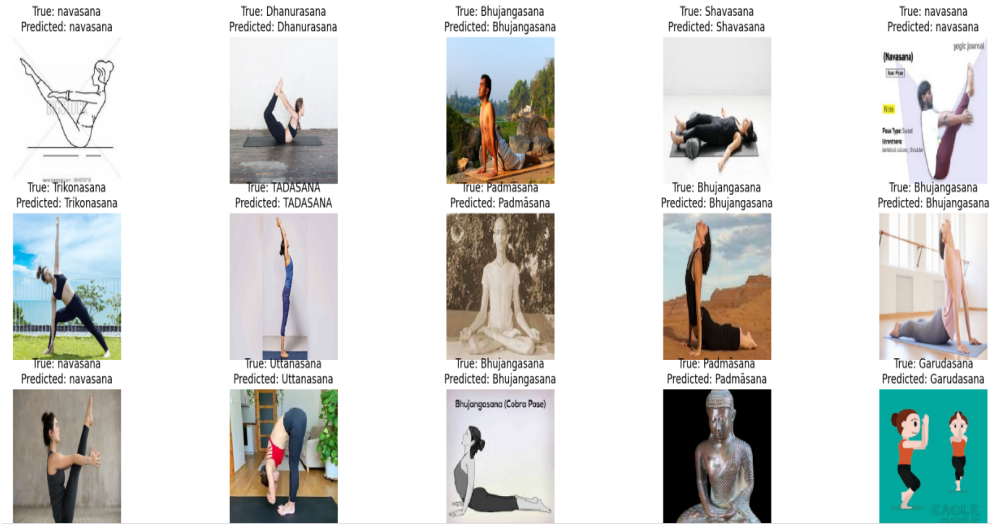

Fig. 9: Predictions on Test Data

The predictions from test set is depicted in Figure 9. We have selected fifteen images randomly from the test set and tested their predictions. Here, all the images of the yoga pose are correctly predicted, as seen in Figure 9.

5 Limitations and Challenges

This section outlines the challenges encountered during experimentation, offering a more balanced and thorough perspective.

Data Availability and Quality: Limited availability of diverse labeled datasets posed a challenge in achieving higher generalization across different Yogasana poses. The quality and variability of the data also influenced model performance.

Computational Complexity: The integration of the self-attention mechanism with the Xception model increased computational demands, requiring optimized hardware resources and efficient implementation strategies.

Pose Variability: The inherent variations in Yogasana postures across individuals, including differences in body alignment and flexibility, introduced challenges in ensuring consistent classification accuracy.

Augmentation Effects: While data augmentation helped improve model generalization, careful selection of augmentation techniques was necessary to avoid introducing artificial biases.

6 Conclusion

Advances and discoveries in science and engineering pave the way for the exploration of new possibilities in transdisciplinary sectors. Modern technologies like artificial intelligence, machine learning, and deep learning are used in many real-time applications in

our daily lives.Yogasana practice considerably improves flexibility, strength, and balance. We demonstrated a machine-learning yoga pose identification system that takes images as a starting point to create a system that can help humans practise yoga with an online trainer. We tried with cutting-edge image classification techniques like as CNN when constructing this system, however owing to the little amount of data in the dataset, such methods did not perform effectively. We used transfer learning with many architectures like DenseNet121, InceptionV3, MobileNet, and ResNet with different layers like attention and CBAM to get better results, but the results with Xception with the self attention network were quite promising; it gave a 99% accuracy which outperforms all the state-of-the-art models.

Declarations

- **Funding** The manuscript submitted by the authors was not endorsed by any organization.
- **Conflict of interest/Competing interests** Considering the subject matter of this article, the authors do not have conflicts of interest to disclose
- **Availability of data and materials** The dataset has been created by the authors and has already been uploaded to Kaggle. It's now Open-Source datasets for all the researchers. Anyone can download it from the following link https://www.kaggle.com/datasets/aishanikaggle/yogasan-dataset

References

[1] Woodyard, C.: Exploring the therapeutic effects of yoga and its ability to increase quality of life. International journal of yoga **4**(2), 49 (2011)

[2] Ross, A., Touchton-Leonard, K., Yang, L., Wallen, G.: A national survey of yoga instructors and their delivery of yoga therapy. International journal of yoga therapy **26**(1), 83–91 (2016)

[3] Mani, P., Thangavelu, A., Sharma, A., Chaudhari, N.: A real time monitoring system for yoga practitioners. International Journal of Inteligent Engineering & Systems **10**(3), 83–93 (2017)

[4] Du, G., Cao, X., Liang, J., Chen, X., Zhan, Y.: Medical image segmentation based on u-net: A review. Journal of Imaging Science and Technology (2020)

[5] Fatih, B., Kayaalp, F.: Review of machine learning and deep learning models in agriculture. International Advanced Researches and Engineering Journal **5**(2), 309–323 (2021)

[6] Ozbayoglu, A.M., Gudelek, M.U., Sezer, O.B.: Deep learning for financial applications: A survey. Applied Soft Computing **93**, 106384 (2020)

[7] Alzubaidi, L., Zhang, J., Humaidi, A.J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M.A., Al-Amidie, M., Farhan, L.: Review of deep

learning: Concepts, cnn architectures, challenges, applications, future directions. Journal of big Data **8**, 1–74 (2021)

[8] Elngar, A.A., Arafa, M., Fathy, A., Moustafa, B., Mahmoud, O., Shaban, M., Fawzy, N.: Image classification based on cnn: a survey. Journal of Cybersecurity and Information Management **6**(1), 18–50 (2021)

[9] Xiao, Y., Tian, Z., Yu, J., Zhang, Y., Liu, S., Du, S., Lan, X.: A review of object detection based on deep learning. Multimedia Tools and Applications **79**, 23729–23791 (2020)

[10] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)

[11] Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708 (2017)

[12] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826 (2016)

[13] Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1251–1258 (2017)

[14] Rafi, U., Leibe, B., Gall, J., Kostrikov, I.: An efficient convolutional network for human pose estimation. In: BMVC, vol. 1, p. 2 (2016)

[15] Yadav, S.K., Singh, A., Gupta, A., Raheja, J.L.: Real-time yoga recognition using deep learning. Neural Computing and Applications **31**, 9349–9361 (2019)

[16] Kothari, S.: Yoga pose classification using deep learning (2020)

[17] Hu, X., Lin, T.Y., Raghavan, V., Wah, B., Baeza-yates, R., Fox, G., Shahabi, C., Yang, Q.: Proceedings of the 2013 ieee international conference on big data. In: Proceedings of the 2013 IEEE International Conference on Big Data (2013)

[18] Trejo, E.W., Yuan, P.: Recognition of yoga poses through an interactive system with kinect device. In: 2018 2nd International Conference on Robotics and Automation Sciences (ICRAS), pp. 1–5 (2018). IEEE

[19] Jain, S., Rustagi, A., Saurav, S., Saini, R., Singh, S.: Three-dimensional cnn-inspired deep learning architecture for yoga pose recognition in the real-world environment. Neural Computing and Applications **33**, 6427–6441 (2021)

[20] Yadav, S.K., Singh, A., Gupta, A., Raheja, J.L.: Real-time yoga recognition using

deep learning. Neural Computing and Applications **31**, 9349–9361 (2019)

[21] Shailesh, S., Judy, M.: Automatic annotation of dance videos based on foot postures. Indian Journal of Computer Science And Engineering, Engg Journals Publications-ISSN **976**, 5166 (2020)

[22] Shailesh, S., Judy, M.: Computational framework with novel features for classification of foot postures in indian classical dance. Intelligent Decision Technologies **14**(1), 119–132 (2020)

[23] Dantone, M., Gall, J., Leistner, C., Van Gool, L.: Human pose estimation using body parts dependent joint regressors. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3041–3048 (2013)

[24] Haque, S., Rabby, A.S.A., Laboni, M.A., Neehal, N., Hossain, S.A.: Exnet: deep neural network for exercise pose detection. In: Recent Trends in Image Processing and Pattern Recognition: Second International Conference, RTIP2R 2018, Solapur, India, December 21–22, 2018, Revised Selected Papers, Part I 2, pp. 186–193 (2019). Springer

[25] Mehta, D., Sotnychenko, O., Mueller, F., Xu, W., Elgharib, M., Fua, P., Seidel, H.-P., Rhodin, H., Pons-Moll, G., Theobalt, C.: Xnect: Real-time multi-person 3d motion capture with a single rgb camera. Acm Transactions On Graphics (TOG) **39**(4), 82–1 (2020)

[26] Jose, J., Shailesh, S.: Yoga asana identification: a deep learning approach. In: IOP Conference Series: Materials Science and Engineering, vol. 1110, p. 012002 (2021). IOP Publishing

[27] Garg, S., Saxena, A., Gupta, R.: Yoga pose classification: a cnn and mediapipe inspired deep learning approach for real-world application. Journal of Ambient Intelligence and Humanized Computing **14**(12), 16551–16562 (2023)

[28] Dhanyal, S.S., Nandyal, S.S.: Yoga pose annotation and classification by using time-distributed convolutional neural network. Indonesian Journal of Electrical Engineering and Computer Science **32**(3), 1639–1647 (2023)

[29] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)

[30] Liu, Y., Zhang, L., Hao, Z., Yang, Z., Wang, S., Zhou, X., Chang, Q.: An xception model based on residual attention mechanism for the classification of benign and malignant gastric ulcers. Scientific Reports **12**(1), 15365 (2022)

[31] Hoogi, A., Wilcox, B., Gupta, Y., Rubin, D.L.: Self-attention capsule networks for object classification. arXiv preprint arXiv:1904.12483 (2019)